UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/633,113	08/01/2003	Michael Beuten	10191/3300	3639

26646 7590 06/13/2011

KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER	
CHERY, MARDOCHEE	

ART UNIT	PAPER NUMBER
2188	

MAIL DATE	DELIVERY MODE
06/13/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MICHAEL BEUTEN, KLAUS SCHNEIDER,
MATTHIAS KNAUSS, and PETER POINSTINGL

Appeal 2009-010164
Application 10/633,113
Technology Center 2100

Before JOHN A. JEFFERY, THU A. DANG, and DENISE M. POTHIER, Administrative Patent Judges.

POTHIER, Administrative Patent Judge.

DECISION ON APPEAL

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-10 and 12. Claim 11 has been canceled. App. Br. 2. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

STATEMENT OF THE CASE

Appellant's invention manages dynamic memory without having to change the startup code. See generally Spec. 2, 6. Claim 1 is reproduced below:

1. A method for providing dynamic memory management of a memory device, the method comprising:
providing a first memory block in the memory device;
storing a startup program in the first memory block;
providing additional memory blocks; and
connecting the first memory block and the additional memory blocks by a chained list;
wherein the memory device is checked before the chained list is executed and the startup program obtains data for a check from the additional memory blocks.

The Examiner relies on the following as evidence of unpatentability:

Sorber	US 6,088,777	July 11, 2000
Bright	US 6,141,756	Oct. 31, 2000
Porterfield	US 6,192,457 B1	Feb. 20, 2001

THE REJECTION

The Examiner rejected claims 1-10 and 12 under 35 U.S.C. § 103(a) as unpatentable over Sorber, Porterfield, and Bright. Ans. 3-6.[1]

THE CONTENTIONS

Regarding representative independent claim 1, the Examiner finds that Sorber discloses all recited limitations, except for checking the memory device before the chained list is executed and the startup program obtains data for a check from the additional memory blocks. Ans. 3-4. The Examiner cites: (1) Porterfield to teach a startup program that obtains data

[1] Throughout this opinion, we refer to (1) the Appeal Brief filed September 7, 2007; (2) the Examiner's Answer mailed November 9, 2007; and (3) the Reply Brief filed January 11, 2008.

for a check from additional memory blocks when combined with Sorber (Ans. 4), and (2) Bright to teach checking a device before executing when combined with Sorber (Ans. 4-5).

Appellants argue: (1) Sorber does not disclose storing a startup program in a first memory block and connecting the first memory block to additional memory blocks in a chained list because memory 20 and 22 are separately located; (2) Porterfield does not relate to a startup program obtaining data for a check from the additional blocks; (3) Bright does not suggest a memory device is checked before the chained list is executed; and (4) Sorber, Porterfield, and Bright are unrelated and there is no reason to combine these references. App. Br. 6-10; Reply Br. 3-7.

ISSUES

(1) Under § 103, has the Examiner erred in rejecting claim 1 by finding that Sorber, Porterfield, and Bright collectively would have taught or suggested:

(a) storing a startup program in a first memory block of a memory device;

(b) connecting the first memory block and the additional memory blocks by a chained list;

(c) checking the memory device before the chained list is executed; and

(d) the startup program obtaining data for a check from the additional memory blocks?

(2) Is the Examiner's reason to combine the teachings of Sorber, Porterfield, and Bright supported by articulated reasoning with some rational underpinning to justify the Examiner's obviousness conclusion?

FINDINGS OF FACT (FF)

1. Appellants state in the Background section that "[a] start program or startup code may be included in the memory device, which is stored in a boot area or BIOS (Basic Input/Output System) and which contains the program instructions necessary when booting up the microcontroller . . . . The functions stored in memory device may be checked starting from the boot block. . . . This occurs, for instance, upon booting the microcontroller . . . ." Spec. 1:10-14, 19-21.

2. Appellants explain a chained list "means that a reference to next memory block 18 arranged in the sequence of the chained list is always stored in each memory block 18." Spec. 4:26-28; 6:17-21; Fig. 1.

3. We adopt the Examiner's finding that: (1) Sorber's discussion of a memory system 10 booting up includes running a startup program from memory, which involves loading the operating system and other application software (Ans. 7; Sorber, col. 7, ll. 15-28; Fig. 2); (2) Sorber discloses the memory manager receives a pointer (see id.); and (3) Sorber discusses linking each memory block to the next memory block and the first memory block is indicated by a pointer (Ans. 8; Sorber, col. 16, ll. 54-57).

4. Sorber discloses a data processing and memory system 10 having (a) a program memory 20 with a memory manager 26, which receives a pointer or starting memory address from the operating system 24, and (b) data memory 22 that is dynamically allocated. Sorber also discusses

memory blocks are linked together with other memory blocks and the memory block may be linked to other blocks in a list (i.e., a string of linked blocks), when the memory block is part of message. Sorber, col. 7, ll. 1-32; col. 8, ll. 45-50, 55-59; Fig. 2.

5. When the computer system 50 is initialized upon being turned ON, Porterfield teaches BIOS software setting the addresses allocated for each computer device in the system address allocation table. Porterfield, col. 3, l. 64 – col. 4, l. 13; Fig. 2.

6. Bright teaches a bootstrap or test mode for processors that downloads a bootstrap program. Bright provides a security feature, such as checksum or hash, which must be authenticated by the processor 101 before the program may be executed. Bright, col. 1, ll. 6-30; col. 3, ll. 10-28; Fig. 1.

ANALYSIS

Based on the record before us, we find no error in the Examiner's obviousness rejection of independent claim 1. As the Examiner explains (Ans. 7), the startup program is mapped to both the boot up process along with loading the operating system and other applications into memory. See FF 3. Such a "boot up" or startup program must be stored within blocks of a system's memory or Sorber's data processing and memory system 10. See FF 4. Otherwise, the startup program could not be executed. Sorber, therefore, discloses storing a startup program in a first memory block as recited in claim 1.

Additionally, Appellants admit in the Background section of the disclosure that a startup program, which includes BIOS, is stored in a

memory device and contains the instructions for booting up a processor. See FF 1. Thus, the Examiner's discussion of BIOS in Porterfield (see Ans. 4) illustrates a known bootup or startup program used with a memory system when a processing system is initially turned on and initialized. See FF 1, 5. We therefore agree with the Examiner (Ans. 9) that BIOS is recognized by ordinarily skilled artisans to include routines at startup and take notice that tests or checks on hardware, including memory, using the BIOS routines are well known.[2] Combining Sorber with Porterfield further teaches and suggests that memory devices, such as Sorber's, are checked before connecting any hardware (e.g., memory blocks), including before executing the chained list as recited in claim 1. Bright additionally supports this conclusion, by demonstrating a technique for performing a test or check on a processing system before executing an operating program. See FF 6. Sorber, Porterfield, and Bright collectively therefore teach or suggest checking a memory device before executing a chained list, as recited by claim 1.

We also take notice of a known test performed by BIOS or the Power-on self-test (POST) on processing devices, which finds and verifies the system's memory. Accounting for inferences and creative steps that an ordinarily skilled artisan would have employed, we find that such BIOS testing would involve sending and receiving data from all tested memory or memory blocks, such as Sorber's additional data memory blocks 22 (see FF 4), so as to find and verify the system's memory. See *KSR Int'l Co. v.*

[2] See In re Ahlert, 424 F.2d 1088, 1091 (CCPA 1970) (explaining that "the Patent Office appellate tribunals, where it is found necessary, may take notice of facts beyond the record which, while not generally notorious, are capable of such instant and unquestionable demonstration to defy dispute.")

Teleflex, Inc., 550 U.S. 398, 418 (2007). As such, the taught BIOS startup program would involve obtaining some information or data from memory blocks, including the recited additional memory blocks, so as to verify or check the system's memory.

Regarding the limitation, "connecting the first memory block and the additional memory blocks by a chained list," Appellants argue Sorber's first memory block (e.g., 20) is not connected to the additional memory blocks (App. Br. 6; Reply Br. 3-4) and that Sorber's teaching of the first memory block does not refer to the memory block that stores the startup program (App. Br. 6-7; Reply Br. 3-5). We disagree.

Appellants explain a chained list includes each memory block containing a reference to the next memory block in a sequence of the chained list. See FF 2. Thus, when read in light of the disclosure, the recitation of connecting the first and additional memory blocks by a chained list includes having a first memory block that references the next memory block in a sequence (e.g., an additional memory block) and the next memory block references yet another memory block (e.g., yet another additional memory block). See id. As the Examiner notes (see FF3), Sorber discusses linking or referencing each additional memory block (e.g., 32) to the next memory block (see FF 4) and also that the operating system stored in memory (e.g., a first memory block) contains a pointer to the starting memory address (see id.). Thus, Sorber teaches connecting a memory block that stores a program installed by the startup program (e.g., the operating system) and is also part of the startup procedure to additional memory blocks that are also linked. See FF 3-4. We therefore find that Sorber

teaches and suggests connecting the first memory block and the additional memory blocks by a chained list as recited.

Lastly, because the cited references are purportedly unrelated, Appellants assert that the Examiner has not provided an adequate reason with rational underpinning to combine the references as suggested by the Examiner. App. Br. 10; Reply Br. 7. Porterfield is cited for the limited purpose of teaching that BIOS, a startup program, runs when a memory system is turned on and during initialization. See FF 5. When the startup program operates is reasonably pertinent the problem with which the inventor was concerned (see FF 1; see also claim 1) and is thus analogous art. See In re Kahn, 441 F.3d 977, 986-87 (Fed. Cir. 2006). Additionally, Bright is reasonably pertinent to when the startup program executes a check on the system and also analogous art. See FF 6; see also claim 1. Moreover, the Examiner has articulated some reasoning with some rational underpinning to combine these references as suggested. See Ans. 4-5, 9-12. We therefore find the references are analogous for the limited purpose for which they are cited and some reason with a rational underpinning to combine the references has been articulated by the Examiner. See KSR, 550 U.S. at 418.

For the foregoing reasons, Appellants have not persuaded us of error in the obviousness rejection of: independent claim 1 and claims 2-10 and 12 not separately argued with particularity (App. Br. 4-10).

CONCLUSION

The Examiner did not err in rejecting claims 1-10 and 12 under § 103.

DECISION

The Examiner's decision rejecting claims 1-10 and 12 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

ELD